Pencarian Rute Terdekat Pada Labirin Menggunakan Metode A*

Rengga Dionata Putra, Ir. Muhammad Aswin, MT. Dan **Waru Djuriatno, ST., MT.**

Abtrak - Labirin adalah sebuah jaringan dari jalur jalur yang saling berhubungan untuk dilalui dari awal hingga akhir yang dimaksudkan untuk sebuah tantangan, manusia mungkin masih dapat menyelesaikan masalah pencarian ruang terdekat yang sederhana, tetapi jika jumlah rute yang ada sudah sedemikian banyaknya, maka kita akan mengalami kesulitan dan akan memakan waktu yang lama untuk menyelesaikannya.

I. PENDAHULUAN

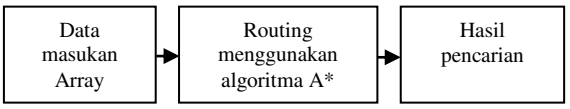
Pencarian rute terdekat, adalah usaha untuk mencari rute yang paling dekat dari posisi awal hingga akhir dengan beban paling ringan atau sedikit dibandingkan dengan seluruh rute yang ada.

Salah satu lingkungan yang banyak digunakan manusia sebagai permainan yang juga merupakan salah satu bentuk lingkungan dari pencarian rute terdekat yaitu adalah labirin. Labirin merupakan sebuah ruang yang memiliki banyak jalur dan persimpangan, dan pemain harus menemukan rute terdekat dari posisi awal hingga posisi akhir

Algoritma A* adalah sebuah algoritma yang telah diperkaya dengan menerapkan suatu *heuristik*, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan

II. RANCANGAN PENELITIAN

A. Blok Diagram Keseluruhan Sistem



Gambar 1. Blok Diagram Keseluruhan Sistem

Blok diagram keseluruhan sistem dari penelitian ini seperti pada Gambar 1, dimana data masukan berupa array yang akan visualisasikan terlebih dahulu menjadi labirin yang selanjunya akan di hitung langkahnya sampai menemukan garis finish.

*B. A**

Algoritma A* menyelesaikan masalah yang menggunakan graf untuk perluasan ruang statusnya. Dengan menerapkan suatu heuristik. Heuristik adalah nilai yang memberi nilai pada tiap simpul yang memandu A* mendapatkan solusi yang diinginkan. Dengan heuristik, maka A* pasti akan mendapatkan solusi (jika memang ada solusinya). Dengan kata lain, heuristik adalah fungsi optimasi yang menjadikan algoritma A* lebih baik dari pada algoritma lainnya.

C. Djikstra

Algoritma ini adalah salah satu bentuk algoritma greedy. Padsa algoritma ini pemeriksaan simpul akan dilakukan ke segala arah yang dimungkinkan yang pada akhirnya seluruh simpul pada sebuah graf akan diperiksa. Hal ini menyebabkan algoritma ini bekerja dengan lambat dan menggunakan memori yang besar sehingga waktu yang diperlukan untuk menemukan solusi akan semakin besar pula.

D. Routing

Routing/Perutean proses untuk memilih jalur (path) yang harus dilalui oleh paket. Semua routing bertujuan mencari rute tersingkat untuk mencapai tujuan. Dan masing-masing protokol mempunyai cara dan metodenya sendiri-sendiri.

Pada aplikasi yang di buat agen harus bisa mencapai garis finish sebagai alamat akhir yang harus dituju, dengan proses pemilihan dan perhitungan jalur pada setial langkahnya

III. PERANCANGAN DAN IMLEMENTASI

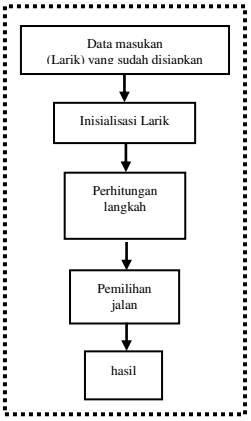
Perancangan ini meliputi pembuatan lunak, pembuatan program pada visual basic untuk keperluan perhitungan dan pemilihan langkah hingga menemukan garis finih.

A. Perancangan PerangkatLlunak

Perangkat lunak yang akan dibuat menggunakan bahasa pemrogaman Visual Basic.NET 2010 dan sistem yang digunakan untuk membangun perangkat lunak ini dirancang dengan spesifikasi mampu melakukan hal-hal sebagai berikut:

1. merubah data masukkan yang berupa Larik menjadi bentuk labirin
2. menghitung dan mencari jalan tercepat pada labirin

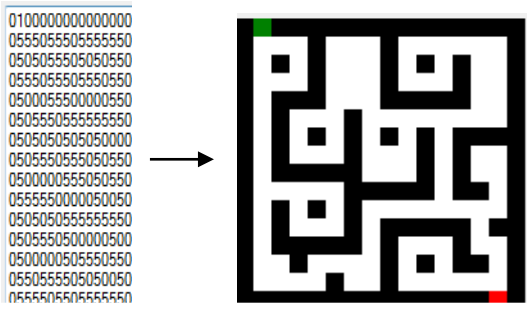
B. desain aplikasi



Langkh pertama adalah menyiapkan larik sebagai inputan utama, setelah itu aplikasi akan memvisuaisasikan menjadi labirin untuk dihitung dan memili langkah hingga finis.

C. *Inisialisasi larik*

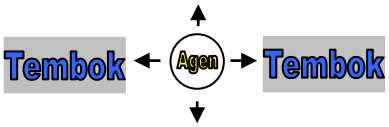
Pada tahap ini aplikasi akan menginisialisasi data masukan yang kita persiapkan sebelumnya, menjadi sebuah labirin yang utuh, proses yang teradi pada tahap ini adalah, membedakan kode tembok, kode jalan masuk, kode jalan keluar dan kode jalan.



Gambar1. visualisasi data larik menjadi labirin

D. *Pemilihan langkah*

Pada tahap pemilihan langkah ini, dilakukan setelah data Larik di inisialisasi dan menjadi labirin secara utuh. Proses ini dilakukan dengan cara membandingkan area sekitar agen ter lebih dahulu, yaitu : area atas, bawah, kanan dan kiri labirin. agen akan menginisialisasi area sekitar dan membandingkan antara tembok, jalan ataukah jalan buntu

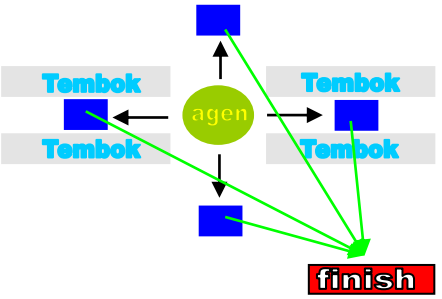


Gamba 2. inisialisasi agen terhadp area sekitar

E. *Perhitungn Langkah*

Perhitungan akan langkah dilakukan ketika agen berada pada posisi awal (jalan masuk) hal ini bertujuan agar agen sudah bisa memilih langkah pada awal posisi. Pada aplikasi pencari rute terdekat ini menggunakan pytagoras untuk menghitung jarak.

Perhitungan jarak dilakukan dengan menghitung jarak atar koordinat saat ini (x,y) dan finish (x,y). Perhitungan terus dilakukan ketika agen bergerak hingga agen menemukan jalan keluar/finish

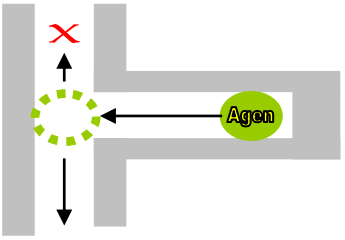


Gambar 3. perhitungan area sekitar agen

Pada gambar diatas, agen menghitung area sekitar meliputi atas, bawah, kanan, kiri (bila memungkinkan) dengangaris finish menggunakanmetde pytagoras, sehingga didapat hasil obt yang paling ringan.

F. *Back track*

Proses backtrack ini hanya digunakan untuk labirin yang mempunyai jalan buntu atau rumit sehingga agen bisa bergerak mundur dan memilih jalan yang memberikan pilihan untuk jalan.



Gambar 4. Proses backtrack

Pada gambar diatas, agen beada pada jalan buntu da akan kembali ke persimpangan yang memberi pilihan untuk jalan. Tanda “X” berarti jalan suah dilalui dan tak boleh dilalui oleh agen lagi sehingga agen akan bergerak kebawah.

IV. PENGUJIAN

Uji coba ini adalah cara untuk mengetahui hasil dari percobaan yang dilakukan sekaligus sebagi sarana pemunculan ide ide bagi proses pengembangan selanjutnya. Pengujian yang dilakukan dalam bab ini adalah sebagai berikut :

1. labirin tanpa cabang
2. labirin bercabang
3. labirin buntu bersolusi
4. labirin buntu
5. labirin 20 x 20 bercabang, buntu bersolusi

A. *Labirin tanpa cabang*

labirin yang hanya mempunyai satu jalan atau labirin yang tidak memberikan agen pilihan jalan untuk dipilih.

Gambar5. pengujian labiin tanpa c abang

Pada pengujian yang dilakukan di labirin tidak bercabang ini, agen berhasil menemukan jalan keluar/finish dengan 29 langkah.

B. Labirin Bercabang

Labirin bercabang adalah labirin yang mempunyai jalan lebih dari satu, sehingga ada beberapa kemungkinan mencapai jalan keluar, agen pun diberikan pilihan untuk memilih jalan

Gambar 6. Labirin Bercabang

Pada pengujian labirin bercabang ini agen memerlukan 56 langkah untuk mencapai jalan keluar

C. Labirin Buntu Bersolusi

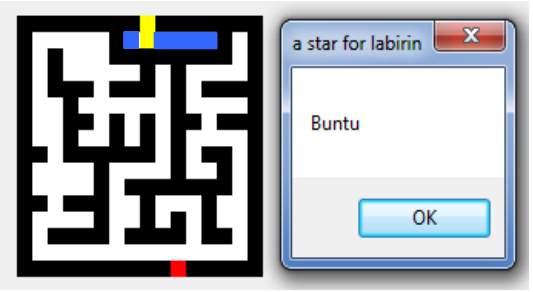
Labirin buntu bersolusi adalah labirin yang mempunyai jalan buntu, bercabang tapi tetap mempunyai solusi, dalam artian agen masih bisa mencapai jalan keluar/finish

Gambar 7. Labirin Bntu Bersolusi

Pada pengujian pertama pada labirin buntu bersolusi ini agen memerlukan 36 langkah untuk menemukan jalan keluar, pada gambar labirin diatas agen melakukan proses backtrack, proses ini di tandai dengan jalan berwarna biru pada gambar

D. Labiri Buntu

Labirin buntu adalah abirin yang tak mempunyai jalan keluar.

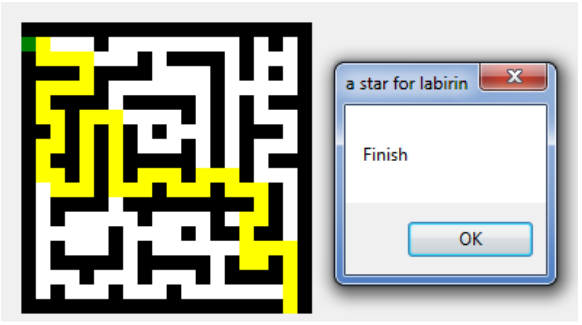


Gambar 8. Labirin buntu

Pada pengujian pertama ini agen melakukan backtrack, dan memerlukan 12 langkah untuk kembali ke posisi awal, karena tak mendapatkan solusi.

E. Labirin 20 x 20 Bercabang, Buntu Bersolusi

Labirin 20 x 20 adalah labirin yang mempunyai panjang 20 kotak dan lebar 20 kotak. Labirin ini mempunyai percabangan dan jalan buntu sehingga lebih rumit dari pengujian pengujian sebelumnya.



Gambar 8. Labirin 20x20 Bercabang dan buntu bersolusi

agen melakukan 62 langkah untuk menemukan jalan keluar

V. Kesimpulan & saran

Berdarsarkn hasil perancangan, implementasi dan pengujian yang dilakukan, didapatkan kesimpulan :

A* tidak menjamin selalu mendapatkan rute terbaik (bobot terkecil) dari semua rute yang ada, hal ini terjadi karena A* hanya menghitung area yang dilalui saja. Area yang tidak dilalui di abaikan

Saran yang dapat diberikan untuk pengembangan pencarian rute terdekat pada labirin menggunakan metode A* antara lain :

1. untuk pengembangan lebih lanjut pencarian rute terdekat ini, menggunakan labirin yang lebih besar dan lebih rumit
2. ditambahkan kriteria atau jenis jenis labirin yang lebih kompleks

DAFTAR PUSTAKA

1. Kusumadewi, Ida dan Hari Purnomo. 2005. *Penyelesaian Masalah Optimisasi dengan Teknik-teknik Heuristik.* Yogyakarta : Graha Ilmu

2. Pratama, Rian Putra. 2011. *Perbandingan Algoritma A* dan Dijkstra Berbasis Web GIS*

untuk Pencarian Rute Terpendek. Skripsi. Bandung: UPI

3. Rudy Adipranata, et al. 2007. Aplikasi Pencari Rute Optimum pada Peta Guna Meningkatkan Efisiensi Waktu Tempuh Pengguna Jalan dengan Metode A* dan Best First Search

4. Aini, Dewi Yusra. 2011. Analisis Algoritma A Star (A*) dan Implementasinya dalam Pencarian Jalur Terpendek pada Jalur Lintas Sumatera di Provinsi Sumatera Utara. Skripsi.

5. Felzenszwalb, Pedro F and D. McAllester. 2007. *The Generalized A* Architecture*. Journal Of Artificial Intelegence Research : hal. 153-190.

6. Harlianto Tanudjaja, et al.2008.*Sistem Perencanaan Jalur Dengan Metode A* (A Star) pada Perangkat Bantu Tuna Netra*

7. Catatanteknisi.com, pengertian routing, table routing dan protocol routing. www.catatanteknisi.com/2011/05/pengertian-routing-tabel-routing.html#.USCRJ_LVeG8

8. WIKIPEDIA. 2012. COMPLEXITY. WIKIPEDIA.ORG/WIKI/A*_SEARCH_ALGORITHM#COMPLEXITY

9. Algorithmist. 2012. Dijkstra's_algorithm. algorithmist.com/index.php /Dijkstra%27 s_ algorithm

10. A*_SEARCH_ALGORITHM. 2012. HTTP://EN.WIKIPEDIA.ORG/WIKI/A* SEARCH ALGORITHM

11. Thesolidsnake. 2012. http://thesolidsnake.wordpress.com/2012/01/23/pencarian-jarak-terpendek-dengan-algoritma-a/

12. Alihasyim. 2012. pencarian-rute-terpendek-menggunakan. http://alihasyim.blogspot.com/2012/10/pencarian-rute-terpendek-menggunakan.html